Project 2: What's the Password?
Due: Monday, October 30, 2017 at 11:59pm

Description
Throughout most of your CS or CoE studies, you work creating or modifying programs or computers – in a word: building. However, sometimes the best way to learn about something is to break it. In this project you will be deconstructing existing programs that each have a secret password or passphrase that needs to be input in order to unlock the program.

I am providing you with 3 compiled executables. Each one requires you to enter a sequence of ASCII characters to "unlock." Unlocking the programs will draw upon the things we are studying this term.

You will also write a tool to help you with solving the first program. In UNIX/Linux, there is a program called `strings` that dumps out the sequences of ASCII characters that are 4 or more characters long.

Part 1: mystrings (40 points)
The `mystrings` program should take a filename from the command line and read the bytes of the file, looking for strings of printable characters. Printable characters are ASCII values between 32 and 126 decimal and the ASCII value 9 (the tab character). A string is a run of at least 4 consecutive printable characters and ends whenever a non-printable character is encountered. Whenever you find such a string, print it out, one per line.

You can check the operation of your program via the real `strings` program, and do a `man strings` to learn about how it works. The output from your `mystrings` program should match exactly the output from "`strings -a <filename>`". Try comparing the output from object files (.o), image files (.jpg), and text files (.c). The output should be the same regardless of file type.

Make sure your program can handle strings that are arbitrarily long.

Part 2: Passwords (60 points)
For each of the three programs, you will be required to provide two things: the solution passphrase and a written description of your attempts to discover it, stating what you learned to help you along the way. You should relate your experiences back to the course material, using the terms and concepts we've discussed. Write it up in a formal, organized fashion. You do not need to describe every command you have tried or every wrong idea. Describe briefly your failed attempts and motivation, but describe in detail your successful approach.

Tools

The most obvious tool you will need is a good debugger like gdb. You may also find a hex viewer like od -x useful. objdump can do a lot of individual tasks that can be helpful. Additionally, you might find the mystrings command you wrote somewhat useful.

Environment

For this project we will be working on thoth.cs.pitt.edu

When you login via ssh with your Pitt account, you will find a local directory under /u/SysLab/ named with your username. In this directory, you will find the three executables and space to work on them. If you store any files of your own in this directory, note that it is not part of AFS, and only exists on this machine. **We will delete your directory when the term is over.** Anything you want to save or backup should be copied into your AFS private directory.

Hints/Notes

- Each program is written in C
- Each program will have a different passphrase per student, although how to find it will be consistent for everyone
- All passphrases will be printable ASCII characters and be less than 100 characters in length
- A passphrase may be different each run of a program, make sure to test it several times
- There may be several passphrases that work, try to describe them or explain why
- This is not an attempt to prove how clever I am, each program will be solvable from course material and the standard tools on the system.

What to turn in

- Your mystrings program and source code
- A written description for each program documenting your attempts to arrive at the solution and the passphrase itself, submitted as a Word 2003 (.DOC) or PDF document.
- All in a tar.gz file, named with your user id
- **Copy your archive to the appropriate directory:**

```
~wahn/submit/449/RECITATION_CLASS_NUMBER
```